

Research Article

Integrating Biosystem Models Using Waveform Relaxation

Linzhong Li,^{1,2} Robert M. Seymour,^{2,3} and Stephen Baigent²

¹ Institute for Energy Technology (IFE), P.O. Box 40, 2027 Kjeller, Norway

² Department of Mathematics, University College London, Gower Street, London WC1E 6BT, UK

³ Center for Mathematics and Physics in the Life Sciences and Experimental Biology (CoMPLEX), University College London, Wolfson House, 4 Stephenson Way, London NW1 2HE, UK

Correspondence should be addressed to Robert M. Seymour, rms@math.ucl.ac.uk

Received 24 June 2008; Accepted 24 August 2008

Recommended by John Goutsias

Modelling in systems biology often involves the integration of component models into larger composite models. How to do this systematically and efficiently is a significant challenge: coupling of components can be unidirectional or bidirectional, and of variable strengths. We adapt the waveform relaxation (WR) method for parallel computation of ODEs as a general methodology for computing systems of linked submodels. Four test cases are presented: (i) a cascade of unidirectionally and bidirectionally coupled harmonic oscillators, (ii) deterministic and stochastic simulations of calcium oscillations, (iii) single cell calcium oscillations showing complex behaviour such as periodic and chaotic bursting, and (iv) a multicellular calcium model for a cell plate of hepatocytes. We conclude that WR provides a flexible means to deal with multitime-scale computation and model heterogeneity. Global solutions over time can be captured independently of the solution techniques for the individual components, which may be distributed in different computing environments.

Copyright © 2008 Linzhong Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

A component-based methodology is explicitly or implicitly widely applied to the understanding and modelling of biological systems. For example, to represent a cell and its wide range of functions, we have to integrate individual models for relevant metabolic, signalling, and gene expression pathways, as well as the associated biophysical processes for intracellular, intercellular and extracellular transport. At the next scale up, a tissue or organism level model requires the integration of different kinds of cell function and cell-cell communication in their intra- and extracellular environments. This is typical of the bottom-up approach to systems biology, in contrast to the top-down approach, which tends to start from the system as a whole (see [1] for a thorough discussion of such general issues).

Living systems are maintained by a continuous flow of matter and energy, and thus any biological system inevitably will be a subsystem of a larger one. Therefore, the biological modeller typically has to deal with an open, multilevel and multicomponent system, the perceived nature of which evolves with our increasing understanding. A key feature

of such a system is the interactions (or coupling in mathematical terminology) among its heterogeneous components and with the external environment, in which a variety of spatial and temporal scales may exist. These interactions may be strong or weak, unidirectional or multidirectional, depending on the current state of the system, and often generate emergent properties through nonlinear interactions. The diversity of existing modelling techniques adds a further layer of complexity to this situation. Thus models of individual components can be based on different modelling formalisms, such as differential equations, discrete time or discrete event simulations, different levels of abstraction of system behaviours, the extent of available knowledge, and the nature of the phenomena being studied.

It can take many years and enormous effort by many researchers across disciplines to build up a model of a complex biological system, and this only on a coarse-grained level consistent with current understanding, which therefore is constantly in need of refinement as techniques and understanding improve. A general issue, therefore, naturally arises: how do we systematically integrate both existing and well-established, as well as new, or more refined versions of

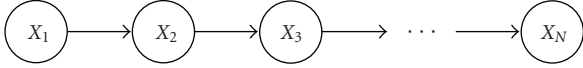


FIGURE 1: A cascade of harmonic oscillators with unidirectional coupling. $X_i = (x_i, y_i)^T$ with $y_i = \dot{x}_i/k_i$.

old, model components in order to build up a larger model system with minimal modification of the internal structure of component submodels?

When describing the behaviour of a complex model system, traditionally we tend to view the system as a whole, implying that the coupling between component parts is implicitly represented. This is driven, in part, by the need to specify suitable mathematical spaces in which whole-system solutions should lie. However, from a computational point of view, it is unnecessary to solve a system as a whole. In contrast to this traditional approach, it is often more natural to construct whole-system behaviours by solving individual components separately, and then to consider the coupling explicitly. This is also often more consistent with developing understanding of the system through the study of separate, isolated components, and makes it possible to update model components individually as knowledge of the detailed biology evolves. Moreover, this approach provides a framework for integrating heterogeneous models (as components of a larger system), which can be distributed in different computational environments.

In the context of integrating biological models, a computational framework under a multicomponent system specification (see [2] for a formal definition) should possess the following features.

- (i) It must be able to represent biological scales both faithfully and economically. This requires a multiscale algorithm, which aims not only to capture the individual biological scales associated with each component but also to resolve the differences of scale between components in a computationally efficient way.
- (ii) The framework should provide the flexibility for integrating models based on different mathematical formalisms, such as deterministic and stochastic simulation. Different mathematical formalisms are often forced upon us by the existence of different spatial or temporal scales.
- (iii) The framework should support encapsulated components. When a new model for a component is developed, we should be able to include it easily, without changing the rest of the framework. This requirement can be termed “plugability”, for example [3].
- (iv) It should also support linking components represented in different software environments, so as to allow new models to be constructed from existing models with minimal changes.

These basic requirements call for a general framework based on a combination of modular, object-oriented design and

agent-oriented design. Modular and object-oriented design provide the flexibility for plugability, and agent-oriented design facilitates the interoperability of coupled models [4]. However, all these designs should be based on a solid theoretical foundation, and also provide a practical means to capture the global dynamic solutions independently of the solution techniques that might be employed for individual components. Suggesting such a methodology is the aim of this work.

Work towards these aims has been published in [3], where a novel algorithm based on a discrete event scheduler was presented in an attempt to meet some of the above requirements. Although successful in many respects, this algorithm is unable to achieve multitime-scale computation for a system with bidirectionally coupled components because of the excessive computational cost of very frequent communication between components. However, bidirectional coupling is common in biological systems due to the ubiquity of feedback mechanisms. For example, in cell signalling, signals are tightly regulated with positive and negative feedbacks that are bidirectional, with commands travelling both from outside-in and inside-out [5].

Here, we offer an approach to deal with bidirectionally coupled components that meets the requirements listed above: waveform relaxation (WR), a flexible numerical method for computing solutions to a system of ordinary differential equations (ODEs) that enables the integration of independently treated subsystems, by using outputs of subsystems as inputs to others and vice versa [6]. When the idea behind the WR method is generalised to a wide range of modelling techniques, it supports a multiscale algorithm that can integrate models of different mathematical forms and provides a way to characterise the dynamical solutions over time and space, independently of the solution technique that might be employed for individual components. However, here we restrict attention to ODE models.

In Section 2, we present a general formulation of the problem of model composition from component models in the context of ODE systems, and in Section 3, the WR method is briefly introduced and generalised, accompanied by some general discussion of how the method is implemented, especially when employed for a discrete event strategy. In Section 4, four case studies are presented to show the capability of the method to cope with different aspects of simulating biological systems. Finally, in Section 5 the results are summarised and discussed.

2. Model Formulation

For simplicity, we consider ODE models to describe briefly our model formulation.

Suppose a given whole-system ODE model S describing a complex biological system. We assume that the state space is an open subset \mathcal{B} of an m -dimensional Euclidean space \mathbb{R}^m , and that the model takes the form of a nonautonomous ODE system:

$$\frac{dY}{dt} = F(Y, t), \quad Y(0) = Y_0 \in \mathcal{B}, \quad t \in [0, T], \quad (1)$$

where $F : \mathcal{B} \times \mathbb{R}_+ \rightarrow \mathbb{R}^m$ satisfies some suitable Lipschitz condition to ensure existence and uniqueness of solutions for $t \in [0, T]$. We assume that S may be decomposed into N component subsystems S_i , $i = 1, \dots, N$, in the following sense. S_i is specified by a state space $\mathcal{B}_i \subseteq \mathbb{R}^{n_i}$ of dimension n_i , together with a linear embedding $\theta_i : \mathbb{R}^{n_i} \subset \mathbb{R}^m$ which expresses state variables for the component S_i as state variables for the composite system S . The set of component systems S_i then forms a *decomposition* of the system S provided that every state variable of S can be expressed as a sum of state variables of component subsystems. More formally, if $Y \in \mathcal{B}$ is a state vector for the whole system, then we can find state vectors $Y_i \in \mathcal{B}_i$ for $i \in I(Y) \subseteq \{1, \dots, N\}$, such that $Y = \sum_{i \in I(Y)} \theta_i(Y_i)$ (this expression need not be unique). This formulation allows the possibility that different components can share some of the same state variables, a device that can facilitate more efficient computation. In what follows, we drop explicit reference to the embeddings θ_i to keep the notation simple.

Using $Y_i \in \mathcal{B}_i$ as a component state vector, and $Y \in \mathcal{B}$ for the whole-system state, we obtain a decomposed model of the form

$$\frac{dY_i}{dt} = F_i(Y, t), \quad Y_i(0) = Y_{i0} \in \mathcal{B}_i, \quad t \in [0, T], \quad i = 1, \dots, N. \quad (2)$$

Now consider the inverse problem of integrating ODE components S_i into a single, composite system S . In this case, we have to supply the functions that define the way in which the components are coupled together.

First, consider an individual (i.e., uncoupled or isolated) ODE component. Let \mathcal{B}_i , \mathcal{P}_i denote subsets of finite-dimensional Euclidean spaces, and let $f_i : \mathcal{B}_i \times \mathcal{P}_i \rightarrow \mathbb{R}^{n_i}$ be a locally Lipschitz function. Then, the i th ODE component is assumed to be specified in the form

$$\frac{dY_i}{dt} = f_i(Y_i, \alpha_i), \quad Y_i(0) = Y_{i0} \in \mathcal{B}_i, \quad t \geq 0. \quad (3)$$

Here, Y_i is the *internal state vector* belonging to the state space \mathcal{B}_i , whose elements are the *internal state variables* associated to the i th component. The second component, α_i , is a supplied time-dependent input that will be used to communicate intercomponent interactions and to represent parameter values, such as internal parameters and external forcing functions (i.e., a vector of *control variables* in the language of control theory). For a completely isolated, autonomous component, the $\alpha_i \in \mathcal{P}_i$ would be just the parameter values needed to run the component alone. However, when not isolated, say as part of an integrated system consisting of N components, the vector function α_i can be further decomposed into two parts, $\alpha_i = (\check{Y}_i, \hat{\alpha}_i)$, one being the *external state vector* \check{Y}_i , whose elements (the *external state variables* of component i) are internal state variables belonging to other components via the intercomponent coupling, and the other being a vector $\hat{\alpha}_i$, representing other internal and external parameters and controls. Thus we assign two types of state variable to each component: internal state variables and external state variables, with the internal

state variables being always state variables of the component independent of whether the component is isolated or not, while the external state variables become state variables (of some other subsystem) only when these components are combined to form a composite system.

For example, in an isolated metabolic system without protein synthesis and degradation, the parameters α_i are the concentrations and the kinetic and binding constants of the enzymes involved, as well as the concentration of external substrates, which are determined by external conditions but not controlled by the system, and can be time dependent. The internal state variables are then the time-dependent concentrations of the intermediary metabolites. When the system is combined with relevant signalling and gene regulatory pathways, parts of the parameters α_i , say the concentrations of the enzymes in the metabolic system, actually become the state variables (i.e., internal state variables) of the signalling and gene regulatory pathways. Thus when we are considering behaviours of the whole system, some parameters originally attributed to these subsystems should not be treated as parameters of the whole system. However, these biologically distinguishable pathways have their disparate time scales, and a variable in one subsystem with a low-relaxation time may be viewed as a bifurcation parameter for the others when the subsystems are isolated.

For such a composed system, once \check{Y}_i , $i = 1, \dots, N$, are specified, a dependence digraph can be constructed, which represents the connectivity of the network, and the bidirectional coupling between components can be rigorously defined in terms of graph theory. Here, for the purpose of representing the WR method in its generic form, we define instead an *influence set* for each i . That is, if $I = \{1, \dots, N\}$ indexes the component subsystems, then $I_i \subseteq I - \{i\}$ is a set of (external) component indices that influence component i . Thus I_i is the collection of indices of those components, some internal state variables of which are the external state variables of component i .

In terms of this conceptualisation, and in order to allow for additional flux transfer between the subsystems modelled by components, we assume that the i th component function f_i in (3) is specified in the form

$$f_i = \hat{f}_i(Y_i, \check{Y}_i, \hat{\alpha}_i) + \varphi_i(Y_i, \check{Y}_i), \quad (4)$$

where the function $\varphi_i(Y_i, \check{Y}_i)$ is reserved for representing flux transfer with other components. It can be constructed from mass action or some other representation, such as Hill functions, of the fluxes due to the interactions of linked components that contribute to the overall flux balance in the whole system. For example, when we write reaction equations using mass action, introducing a new molecular species will result in new flux terms to the original equations, that is, the second term in (4) will appear, while the first term will keep the same meaning as in the original, isolated system.

A composition of the components S_i into a single system S results from a specification of the control variables in $\alpha_i = (\check{Y}_i, \hat{\alpha}_i)$ and the flux transfer function $\varphi_i(Y_i, \check{Y}_i)$. Once these are given, the composed system provides an appropriate set of (time-dependent) functions

$\alpha_i : \mathcal{B} \times \mathbb{R}_+ \rightarrow \mathcal{P}_i$ ($i = 1, \dots, N$). The velocity of Y_i is then given by the functional composition of the supplied component functions f_i and the interaction functions \check{Y}_i . The resulting composed system S is therefore described by the dynamical system (4).

The above formulation suggests that a generic form for any submodel should be provided with the form (4) with a dummy flux term. As an isolated system, this term is set to zero, but as a component in an integrated model, this term can be formed according to flux contributions from the interactions of linked components. The advantage of this formulation is to provide the flexibility to link to other potential models without altering the internal structure of the original model when the WR method, which will be introduced in Section 3, is applied.

3. Computational Approach

In this section, we first provide a brief introduction to the waveform relaxation method (WR), originally developed for the parallel computation of ODEs [6]. We formally generalise the method and discuss some issues relevant to its implementation and efficient computation.

3.1. The Waveform Relaxation Method

We illustrate the WR method using a system decomposed into communicating components. Thus suppose we have a system described by a set of differential equations, decomposed into two subsystems (components) 1 and 2 of the form

$$\begin{aligned} \frac{dY_1}{dt} &= F_1(Y_1, Y_2), \quad Y_1(t_0) = Y_{10}, \\ \frac{dY_2}{dt} &= F_2(Y_1, Y_2), \quad Y_2(t_0) = Y_{20}, \quad t \in [t_0, T], \end{aligned} \quad (5)$$

where $Y_1 = (y_{11}, y_{12}, \dots, y_{1n_1})^T$ and $Y_2 = (y_{21}, y_{22}, \dots, y_{2n_2})^T$ are the state vectors for components 1 and 2, respectively. Then we have two iterative implementation schemes for WR as follows.

(1) Jacobi WR Method

$$\begin{aligned} \frac{dY_1^{(k+1)}}{dt} &= F_1(Y_1^{(k+1)}(t), Y_2^{(k)}(t)), \quad Y_1^{(k+1)}(t_0) = Y_{10}, \\ \frac{dY_2^{(k+1)}}{dt} &= F_2(Y_1^{(k)}(t), Y_2^{(k+1)}(t)), \quad Y_2^{(k+1)}(t_0) = Y_{20}, \end{aligned} \quad (6)$$

for $k = 0, 1, \dots$

(2) Gauss-Seidel WR Method

$$\begin{aligned} \frac{dY_1^{(k+1)}}{dt} &= F_1(Y_1^{(k+1)}(t), Y_2^{(k)}(t)), \quad Y_1^{(k+1)}(t_0) = Y_{10}, \\ \frac{dY_2^{(k+1)}}{dt} &= F_2(Y_1^{(k+1)}(t), Y_2^{(k+1)}(t)), \quad Y_2^{(k+1)}(t_0) = Y_{20}, \end{aligned} \quad (7)$$

for $k = 0, 1, \dots$

Roughly speaking, the Jacobi WR method updates a component based upon the states of all components in the previous iteration, while in the Gauss-Seidel WR method, the new state may also depend on the newly updated states of the current iteration, in addition to states from the previous iteration. The Jacobi and Gauss-Seidel methodologies are widely employed in numerical computation, such as for solving linear or nonlinear algebraic equations and finite difference equations in addition to ODEs. A classical cellular automaton simulation is an example of the Jacobi WR method.

A very important message obtained from the WR method is that a large system can be split into small components, which can be computed independently, while the coupling between components can be realised by an iterative procedure. Conversely, components can be computed independently and by specifying the interfaces between the components and performing an iterative procedure, we are actually simulating a larger system formed by these components. Therefore, we interpret the WR method as an iterative procedure to represent bidirectionally coupled components in such a way that each component can be calculated independently. Of course, if components are unidirectionally coupled, then there is no need to do the iteration and a sequential calculation is sufficient.

Following the notation of Section 2, we can write the WR method in its generic form. For the Jacobi WR method, we have

$$\frac{dY_i^{(k+1)}}{dt} = f_i(Y_i^{(k+1)}(t), \check{Y}_i^{(k)}(t)), \quad Y_i^{(k+1)}(t_0) = Y_{i0}, \quad (8)$$

for $i = 1, 2, k = 0, 1, 2, \dots$, with $\check{Y}_1^{(k)} = Y_2^{(k)}$ and $\check{Y}_2^{(k)} = Y_1^{(k)}$.

In this iterative procedure, the external state variables take their values from the previous iteration and determine the internal state variables in the current iterate. Thus the iteration loop itself does not play a role, since it does not take account of the specific topological structure of the network. Of course, initial guesses for $Y_i^{(0)}$, $i = 1, 2$, have to be given in order to start the iteration.

Much of the complexity of the iterative procedure comes from the Gauss-Seidel WR method, and it should be defined in terms of the dependence digraph to take account of the topological structure of the network. For simplicity, for an N -component system, we assume an iterative loop has been predetermined, say, in the sequential order of $I = \{1, \dots, N\}$ (by relabelling components if necessary), with influence set for the i th component $I_i = \{i_1, i_2, \dots, i_s\}$, where $i_1 < i_2 < \dots < i_s$. The method can be formally defined by

$$\frac{dY_i^{(k+1)}}{dt} = f_i(Y_i^{(k+1)}(t), \check{Y}_i^{(m)}(t)), \quad Y_i^{(k+1)}(t_0) = Y_{i0}, \quad (9)$$

$i = 1, 2, \dots, N, k = 0, 1, 2, \dots$, where

$$\begin{aligned} \check{Y}_i^{(m)}(t) &= (\check{y}_{i_1}^{(m)}, \check{y}_{i_2}^{(m)}, \dots, \check{y}_{i_s}^{(m)})^T, \\ \check{y}_{i_d}^{(m)} &= \begin{cases} \check{y}_{i_d}^{(k+1)}, & i_d < i, \\ \check{y}_{i_d}^{(k)}, & i_d > i. \end{cases} \end{aligned} \quad (10)$$

These two methods are examples of continuous waveform relaxation (continuous referring to the time variable). In a numerical implementation of the method, each set of continuous differential equations has to be discretised into a set of difference equations, and this results in a discrete waveform relaxation. Under fairly general conditions, both continuous and discrete WRs are convergent to the theoretical solutions [6]. Thus we have

$$Y_i^{(k)}(t) \longrightarrow Y_i^*(t) \quad \text{as } k \longrightarrow \infty$$

uniformly for $t \in [0, T]$, $i = 1, 2, \dots, N$,

for any set of initial conditions $Y_i^{(0)} : [0, T] \rightarrow \mathbb{R}^{n_i}$, $i = 1, 2, \dots, N$. Moreover, the limit functions $Y_i^* : [0, T] \rightarrow \mathbb{R}^{n_i}$ satisfy the original system (1). The rate of convergence of the iterates depends on the length T of the time interval on which the iteration is performed, and the way the system is partitioned into subsystems.

Splitting a large system into smaller components has another advantage other than the obvious one for parallel computation. Fast and slow varying components may exist, and these can be solved economically by integrating the slow components using larger step sizes than the fast components, with adaptive step size methods employed to realise this. Otherwise, if we solve the system as a whole, integration must be performed with a single time step, which will be determined by the fastest time scale among all the components.

Although here we present the WR method for a system specified by ODEs, the methodology is applicable to many kinds of system specification. In particular, WR has been generalised to stochastic differential equations (SDEs) [7]. The most widely applied numerical method for simulating stochastic systems is the famous Gillespie method [8]. This method can also be interpreted in terms of WR using Gibson and Bruck's formulation [9]. Specifically, we can treat each reaction as a component of the system of interest, and the iteration procedure in WR can be interpreted as updating states following the dependence graph, since this is a discrete event simulation and there are no simultaneous events; that is, there are no bidirectionally-coupled events. However, such an interpretation is not practically useful without further exploration of efficiency issues.

A more practical application of the WR method lies in simulation for hybrid models, which combine stochastic descriptions (say, for gene expression) and deterministic descriptions (say, for signal transduction). The requirement for such model heterogeneity is common in modelling biological systems [10, 11] and is a significant challenge. In fact, the dynamical systems theory for such hybrid models has been identified as an important future research direction [12]. For such hybrid simulation, the WR method can be applied in a straightforward manner so long as interfaces between deterministic and stochastic submodels can be clearly defined, allowing the WR iteration procedure to be applied to simulate the bidirectional coupling between submodels. In the case studies to be developed in Section 4 below, we will provide examples of this kind of simulation.

The key idea of the WR method is to provide a way to compose a system from its components, or to decompose a system into components, by an explicit specification of the coupling relations among components, independent of the internal specification of these components. The explicit specification of the coupling facilitates an iterative procedure that in one iteration sweeps over the components, updating them based upon the states of other components. In this way, the states of individual components can be computed independently, and their bidirectional communication with states of other components is achieved by this iterative procedure.

Moreover, WR provides a means to compute global solutions over time independently of the solution techniques that might be employed for individual components. Such independence also implies that any multiscale method can be applied to solve components that involve a variety of spatial, as well as temporal scales. Another implication from the WR method is that two different models can be linked together by specifying an interface between them, even without significantly modifying the components (e.g., by adding new flux contributions to the system as in equation (4)) or the solution techniques. That is, WR facilitates model *encapsulation*. This is because communication between components in the implementation of the WR method is carried out by data input and output. Thus when adaptive grid methods both in time and space, stiff solvers, and multirate methods [13] are suitably chosen for all components, a wide range of multiscale computation becomes possible.

3.2. Practical Implementation of WR

In the implementation of WR, the time interval $[t_0, T]$ is partitioned into a set of L subintervals $\{[T_i, T_{i+1}] : i = 0, 1, \dots, L-1, T_0 = t_0, T_L = T\}$. The iteration for bidirectionally coupled components is performed on each subinterval sequentially, that is, starting from $[t_0, T_1]$ and moving to the next after the convergence of the iteration on the current interval is achieved, and so on. This is called a “windowing technique” and its application is necessary to avoid the requirement for excessive storage as well as to reduce the number of iterations. In principle, the subintervals can be chosen adaptively, say representing the largest time scale among all the components in order to maximise the efficiency of a multiscale computation, but here, for illustrative purposes, we suppose that each subinterval is of equal length.

A numerical method is chosen for each component; there is no requirement for the same method to be applied to all components. An interpolation method is also required to facilitate the communication between components through external state variables, which are the inputs to the component under execution. The reason for such a requirement is that when an adaptive method is applied to each component on each iteration, the input values from the external state variables associated to a given component are normally not available. This is because the resulting grid points are not the same for each component and each iteration, and therefore have to be obtained by interpolation for a given component.

As observed earlier, the abstract formulation of a system decomposition in Section 2 allows the possibility of overlapping components; that is, different components sharing some of the same internal state variables. This can potentially result in better convergence properties of the waveform iterates by ameliorating the effects of strong coupling between components [14].

While there is considerable flexibility in the choice of numerical methods for solving individual components and interpolating the solution output, some general rules should be followed in order to obtain high accuracy and efficiency. First, an adaptive step size method should be used to capture the right time scale of each component, and this is where the multitime scale efficiency of WR lies. If models for some components are stiff, then stiff solvers should be used. Second, both the orders of accuracy of integration methods for different components, and the orders of accuracy of the interpolation methods should be consistent so that the accuracy of the whole computation is not lost. In our implementation, we employ both Gear's stiff solver and Prince-Dormand's embedding explicit Runge-Kutta 5(4) [15], and three-point Hermite polynomial interpolation for the differential equation specified system, while Gillespie's method is employed for the stochastic simulation with linear interpolation.

3.3. Monitoring and Utilising Varying Coupling Strengths

Coupling among components is a dynamical property of the system, in the sense that two components bidirectionally coupled together by the specification of a network structure does not mean that the two components are strongly coupled for all time. For example, in the simulation of very large-scale integrated (VLSI) circuits, it was found that strong coupling between components only occurs over short-time intervals [16].

Updating all components even when the coupling between some components is found to be weak would be computationally inefficient. Thus we apply the strategy of discrete event simulation to the WR iteration loop [2]. Specifically, in each iteration loop, before executing any component, we examine the variation between the previous iterate and the present one of both internal and external state variables for that component. If the change of these variables is sufficiently small relative to the value of present iterate, then we skip the calculation of the component. The multitime-scale efficiency of a WR algorithm will be dependent on the computational cost for the iteration, and essentially dependent on the coupling strength among the components. The stronger the coupling, the more iterations are needed. On the other hand, the coupling strength of components is dynamically changeable, and therefore the discrete event strategy proposed actually allows us dynamically to follow the change by adaptively reducing or increasing the number of iterations for each component. In the case studies given below, we will demonstrate this by an example. Further issues related to the application

of the WR method will be discussed along with each case study.

4. Case Studies

We now demonstrate the application of the WR method with four models. The first is a simple cascade of harmonic oscillators discussed in [3]. Although the model is not inspired by a biological system, it serves well as a simple example of the improvement the WR method offers over a standard integration algorithm (i.e., one that does not rely on explicit decomposition and coupling of the model). This model is then modified to include feedback from faster components to slower components giving a bidirectionally coupled system. The second model is based on Höfer's calcium oscillation model [17], reformulated so that a combination of deterministic and stochastic simulation can be executed within the WR framework. The third model is a single-cell calcium model [18] that generates different oscillation patterns ranging from simple periodic oscillations to periodic and chaotic bursting in response to agonist stimulation. The model also has a natural decomposition into 2 modules with distinct time scales. This model therefore provides a suitable test case for validating the convergence property of WR, and the benefits of model decomposition based on time scale differences. Finally, the fourth model, which is also based on Höfer's calcium oscillation model, is a genuinely multicellular model that deals with the synchronisation of calcium oscillations within a plate of heterogeneously coupled hepatocytes.

4.1. A Cascade of Harmonic Oscillators

A system of N unidirectionally, linearly coupled harmonic oscillators is defined by the following equations:

$$\begin{aligned} \frac{d^2 x_1}{dt^2} + k_1^2 x_1 &= 0, \\ \frac{d^2 x_i}{dt^2} + k_i^2 x_i &= k_{i-1} k_i x_{i-1}, \quad i = 2, 3, \dots, N, \end{aligned} \quad (11)$$

where the frequency of the i th harmonic oscillator is $k_i = 2^i$. See Figure 1.

The time scale difference among components is explicit in this model. For instance, the total time scale difference across the frequencies k_i is about 6 orders of magnitude for $N = 20$.

The application of the Gauss-Seidel WR to this unidirectional model results in a sequential execution of each component, and no iteration loop is necessary. This is because the system has triangular structure owing to the unidirectional coupling, and the solution can be obtained sequentially by solving for the i th oscillator as driven by the $(i - 1)$ th oscillator.

The efficiency of this multitime scale computation can be easily understood. We consider the solution on a fixed time interval $[0, T]$. Using a WR method based on an adaptive

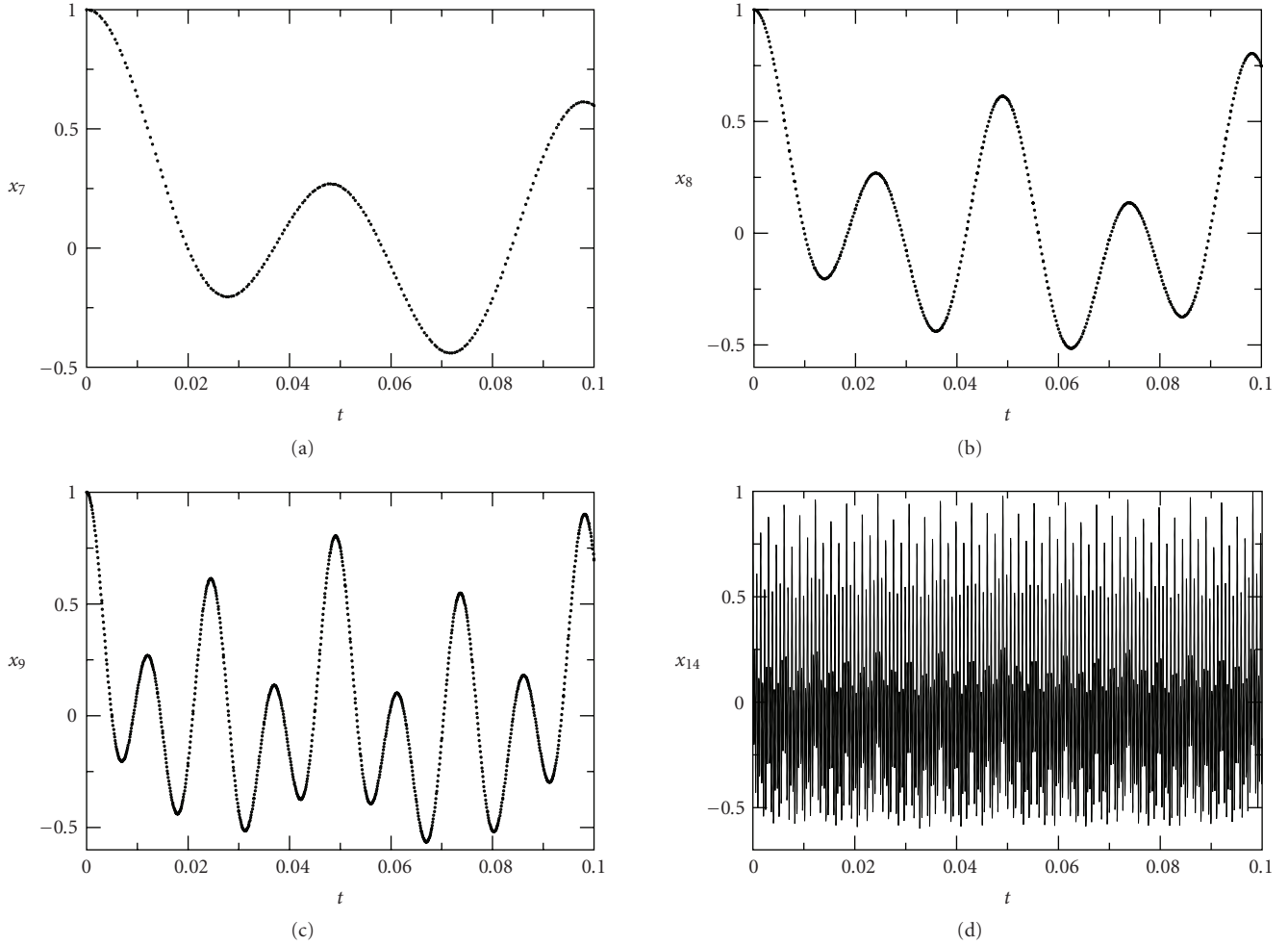


FIGURE 2: Computed amplitudes of units 7, 8, 9, 14 in a cascade of $N = 20$ harmonic oscillators with unidirectional coupling with the initial conditions: $x_i(0) = 0$, $y_i(0) = 1$.

grid method, which precisely follows time-scale variations of component solutions, the number of time steps taken to cover this interval for the i th component is proportional to the frequency k_i . Thus the total number of time steps used for all components is proportional to $k_1 + \dots + k_N = 2(2^N - 1) \approx 2 \times 2^N$ for large N . On the other hand, when a standard adaptive grid method is applied to the whole system, the number of time steps used to reach the end of the interval will be determined by the fastest component (component N in this example), to resolve the variation for this component, regardless of slower variations in other components. That is, the total number of time steps used is proportional to $N \times 2^N$. Thus the WR method is approximately $(1/2)N$ times faster than a standard flat algorithm. In particular, for $N = 20$, the WR method is approximately 10 times faster. This comparison is based on the cost of evaluation of the right-hand functions of the system, with the computational cost of interpolation, which is fixed for a particular interpolation method, neglected. Figure 2 shows the calculated solutions for oscillators 7, 8, 9, 14.

This simple picture changes when we modify the model by adding feedbacks from faster components to their neighbouring slower components (Figure 3), thereby rendering the coupling bidirectional. The ability to cope efficiently with such bidirectional feedback is an important property of any numerical methodology, since feedbacks are important features of biological control systems. This system has no specific biological interpretation. Nevertheless, like the unidirectional system considered above, it provides a significant test case for the WR methodology.

The modified system is given by

$$\begin{aligned} \frac{d^2 x_1}{dt^2} + k_1^2 x_1 &= \varepsilon k_1 k_2 x_2, \\ \frac{d^2 x_i}{dt^2} + k_i^2 x_i &= k_{i-1} k_i x_{i-1} + \varepsilon k_i k_{i+1} x_{i+1}, \quad 2 \leq i \leq N-1, \\ \frac{d^2 x_N}{dt^2} + k_N^2 x_N &= k_{N-1} k_N x_{N-1}. \end{aligned} \quad (12)$$

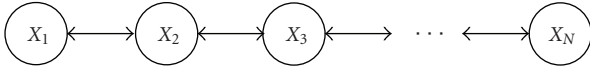


FIGURE 3: The harmonic oscillators with bidirectional coupling. $X_i = (x_i, y_i)^T$ with $y_i = \dot{x}_i/k_i$, $i = 1, 2, \dots, N$.

The parameter ε measures feedback strength. Stability requirements constrain this parameter. For real ε , the stability interval is defined as the range of ε for which the system is composed of N stable harmonic oscillators, and is $(-\infty, \varepsilon_N^*)$, where $\varepsilon_N^* = (1/4)[1 + \tan^2(\pi/(N+1))]$. The detailed stability analysis is given in the Appendix. Interestingly, the stability is independent of specific values of the k 's so long as all of them are positive. Numerical computations done by the Gauss-Seidel WR algorithm verify these stability conditions, as illustrated in Figure 4.

This modified system not only introduces the mutual dependency of neighbouring components but also retains the same multitime-scale character as the original unidirectional system for a suitably chosen ε . Therefore, it provides a model to test the efficiency of the WR algorithm.

First, the computation is performed for $N = 20$ and $\varepsilon = 0.25$, a case with very strong coupling among all neighbouring components. For this computation, there is a tradeoff between the multiscale efficiency and the number of iterations used. The number of iterations will depend on the coupling strength and the length of the time subinterval over which each iteration is performed (i.e., the windowing technique mentioned earlier). For this case, since the coupling is very strong, we have to reduce the subinterval to as small as 10^{-4} to achieve convergence with ten iterations. Here, the convergence criterion is defined by the maximum relative differences between current and previous iterates among all components, which is less than the given error constant 10^{-4} .

Note that there always exists a small subinterval on which the WR iteration is convergent, provided that each submodel satisfies a Lipschitz condition. This can be seen from the proof of WR convergence [6]. However, it is not easy to quantify this subinterval in a general and practical way since it is context dependent. Nevertheless, in real computation this is not a significant issue, since just a few test runs will give an idea about the choice of a suitable subinterval.

However, the analysis of the computational efficiency for the unidirectional coupling above shows that the WR method is approximately 10 times faster than the standard algorithm *on a single interval*. Hence if the method is applied successively more than 10 times, the WR method is no longer efficient compared to the standard algorithm. Therefore, we further reduce the length of subinterval to 10^{-5} . In this case, the average number of iterations required for convergence is 6 and the resulting computational efficiency is comparable with the standard algorithm. Here, we may argue that if components in a system are all coupled very strongly, then separating the system into

components and performing the WR iteration would be not a good choice, and instead multirate methods [13] should be applied to the whole system. Nevertheless, it is a reasonable assumption for a biological system with an identified modular structure to exclude the existence of such strong coupling among components over long-time intervals, since the notion of modularity itself implies strong coupling within components, but weaker coupling between components.

If we reduce the coupling strength ε to 0.01, then a subinterval with length 10^{-4} will result in 5 iterations for convergence, on average. Therefore, for this weaker coupling the algorithm has a better performance than a standard algorithm. In addition, if we assume that the couplings among some of the slower components are strong, but are weak for the remaining faster components, our WR algorithm still gives a better performance than a standard algorithm. Figure 5 shows the computation for this mixed weak and strong coupling. The scenario is arranged by setting $\varepsilon = 0.25$ for the first 10 components ($i = 1, 2, \dots, 10$) and $\varepsilon = 0.001$ for the remaining faster components ($i = 11, 12, \dots, 20$). In this situation, a discrete event-scheduled strategy applied to the iteration loops is very efficient, since it effectively senses the coupling strength and bypasses the components with very small variations.

Notice that all the comparisons above are done in terms of sequential computation, though the WR method has the obvious additional advantage of parallel computation. More precisely, the Jacobi WR method can be directly implemented in a parallel computer, that is, different processes in the computer can deal with different components simultaneously.

Because this system has purely imaginary eigenvalues, some of which have very large magnitude, Gear's method [19] based on backward differentiation is unsuitable. Instead, Prince-Dormand's embedding explicit Runge-Kutta 5(4) [20] was implemented for each component. The computations were compared with the solutions obtained with a single algorithm and agreement is achieved for all the cases discussed above (not shown).

4.2. Nonlinear Oscillators with Nonlinear Coupling in a Calcium Model

In [18], a model for cell calcium dynamics is presented. The main feature of the model is its ability to generate complex oscillations such as periodic bursting and chaotic bursting in response to agonist stimulation, in qualitative agreement with the complex phenomena observed in experiments. The model includes the mechanisms of feedback inhibition on the initial agonist receptor complex by calcium and activated phospholipase C (PLC), as well as receptor type-dependent self-enhanced behaviour of the activated $G\alpha$ subunit. Specifically, let x_1 denote the concentration of active $G\alpha$ subunits, x_2 the concentration of active PLC, x_3 the concentration of free calcium in the cell cytosol, and x_4 the

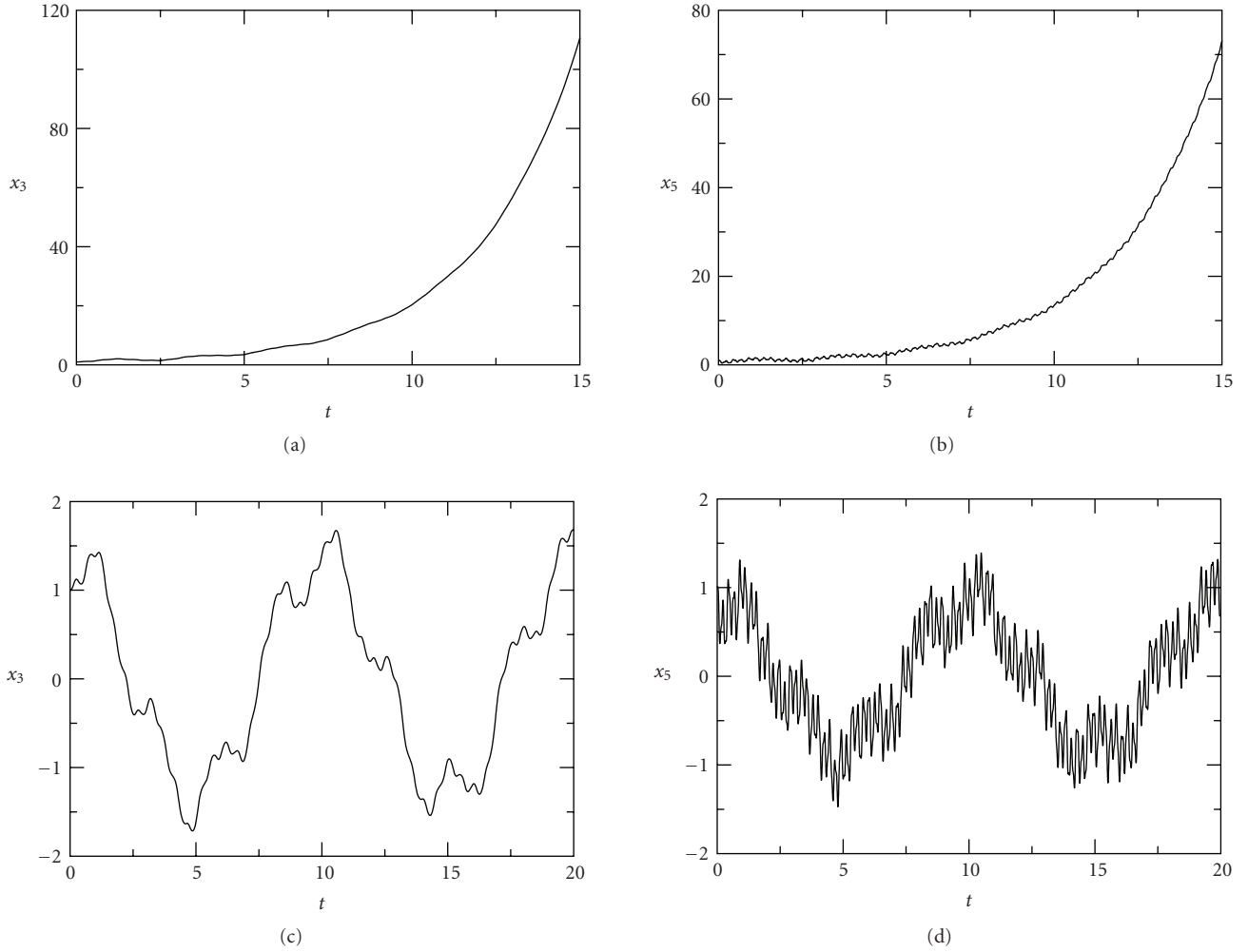


FIGURE 4: The harmonic oscillators with bidirectional coupling. $N = 6$. (a) and (b): $\varepsilon = 0.31$, unstable; (c) and (d): $\varepsilon = 0.30$, stable. $\varepsilon_6^* = (1/4)[1 + \tan^2(\pi/7)] \approx 0.3080$. The initial conditions: $x_i(0) = 0$, $y_i(0) = 1$.

concentration of calcium in the intracellular stores. Then this model is given by the following four nonlinear ODEs:

$$\begin{aligned}
 \frac{dx_1}{dt} &= k_1 + k_2 x_1 - k_3 \frac{x_1 x_2}{x_1 + K_4} - k_5 \frac{x_1 x_3}{x_1 + K_6}, \\
 \frac{dx_2}{dt} &= k_7 x_1 - k_8 \frac{x_2}{x_2 + K_9}, \\
 \frac{dx_3}{dt} &= k_{10} \frac{x_2 x_3 x_4}{x_4 + K_{11}} + k_{12} x_2 + k_{13} x_1 \\
 &\quad - k_{14} \frac{x_3}{x_3 + K_{15}} - k_{16} \frac{x_3}{x_3 + K_{17}}, \\
 \frac{dx_4}{dt} &= -k_{10} \frac{x_2 x_3 x_4}{x_4 + K_{11}} + k_{16} \frac{x_3}{x_3 + K_{17}}.
 \end{aligned} \tag{13}$$

In the component integration approach, a natural question is how do we detect the time-scale differences among state variables so that we can define suitable components each with its own characteristic time scale? Obviously this is

the key for the efficiency of multiscale algorithms. The answer comes from understanding the biology underlying the components. In this calcium model, we expect that the activity changes of $G\alpha$ and PLC in the cell membrane are relatively slow compared with the variation of calcium within the cell [21]. Therefore, we choose to partition the system into just two components, one for $G\alpha$ and PLC and the other for calcium compartments inside the cell, see Figure 6. Then we perform Gauss-Seidel WR iteration for these two coupled components. The computation confirms the supposed large difference in time scales between the two components. For example, in the computation of periodic oscillations the average adaptive step size for the first component is about 0.4 and for the second component is approximately 0.004; that is, two orders of magnitude difference. For the cases with periodic or chaotic bursting, there is still over one order of magnitude difference between the time scales of these two components.

Both Gear's method and Prince-Dormand's method, as well as their combination were applied for the computation,

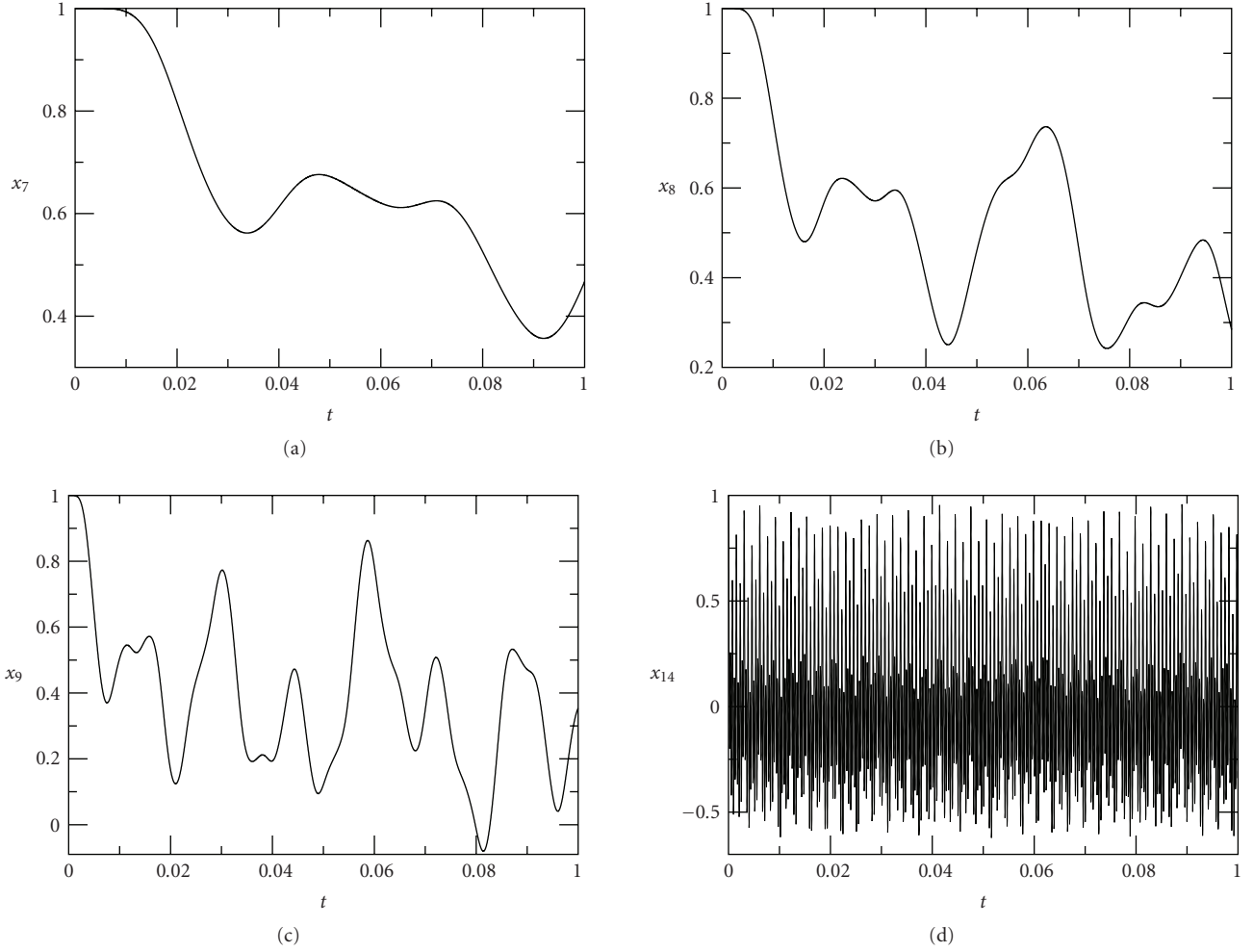


FIGURE 5: The harmonic oscillators bidirectionally coupled with strength $\varepsilon = 0.25$ for components $1, 2, \dots, 10$ and strength $\varepsilon = 0.001$ for components $11, 12, \dots, 20$. The initial conditions: $x_i(0) = 0$, $y_i(0) = 1$.

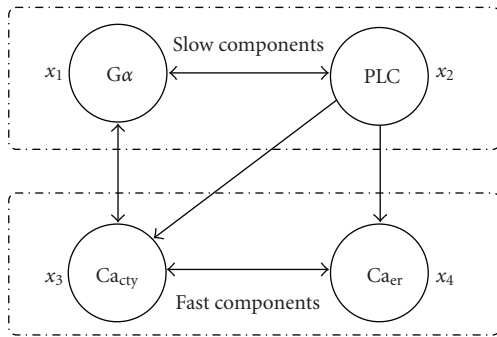


FIGURE 6: Decomposition of Kummer's Calcium model [18].

giving similar performance in terms of convergence, with, on average, 2-3 iterations achieving convergence to within an error constant of 10^{-4} .

This case also indicates that the WR iteration is quite robust even if we have nonlinear oscillators with nonlinear coupling between them and both periodic and chaotic

bursting occur in the solutions. The results, shown in Figure 7, agree qualitatively with the computations done in the original paper [18] with a stiff ODE solver as a single integrator.

4.3. Stochastic/Deterministic Simulations for a Calcium Model

Höfer [17] formulated a calcium model for a hepatocyte, based on flux balances between the endoplasmic reticulum (ER) release (J_{rel}), the ER uptake (J_{SERCA}), the plasma membrane efflux (J_{out}), the calcium influx (J_{in}), and the gap-junctional flux (J_G), resulting in the following two-dimensional system of equations

$$\begin{aligned} \frac{dx}{dt} &= \frac{A_{PM}}{C_C} (J_{in} - J_{out}) + \frac{A_{ER}}{C_C} (J_{rel} - J_{SERCA}) + \frac{A_G}{C_C} J_G, \\ \frac{dz}{dt} &= \frac{A_{PM}}{C_C} (J_{in} - J_{out}) + \frac{A_G}{C_C} J_G, \end{aligned} \quad (14)$$

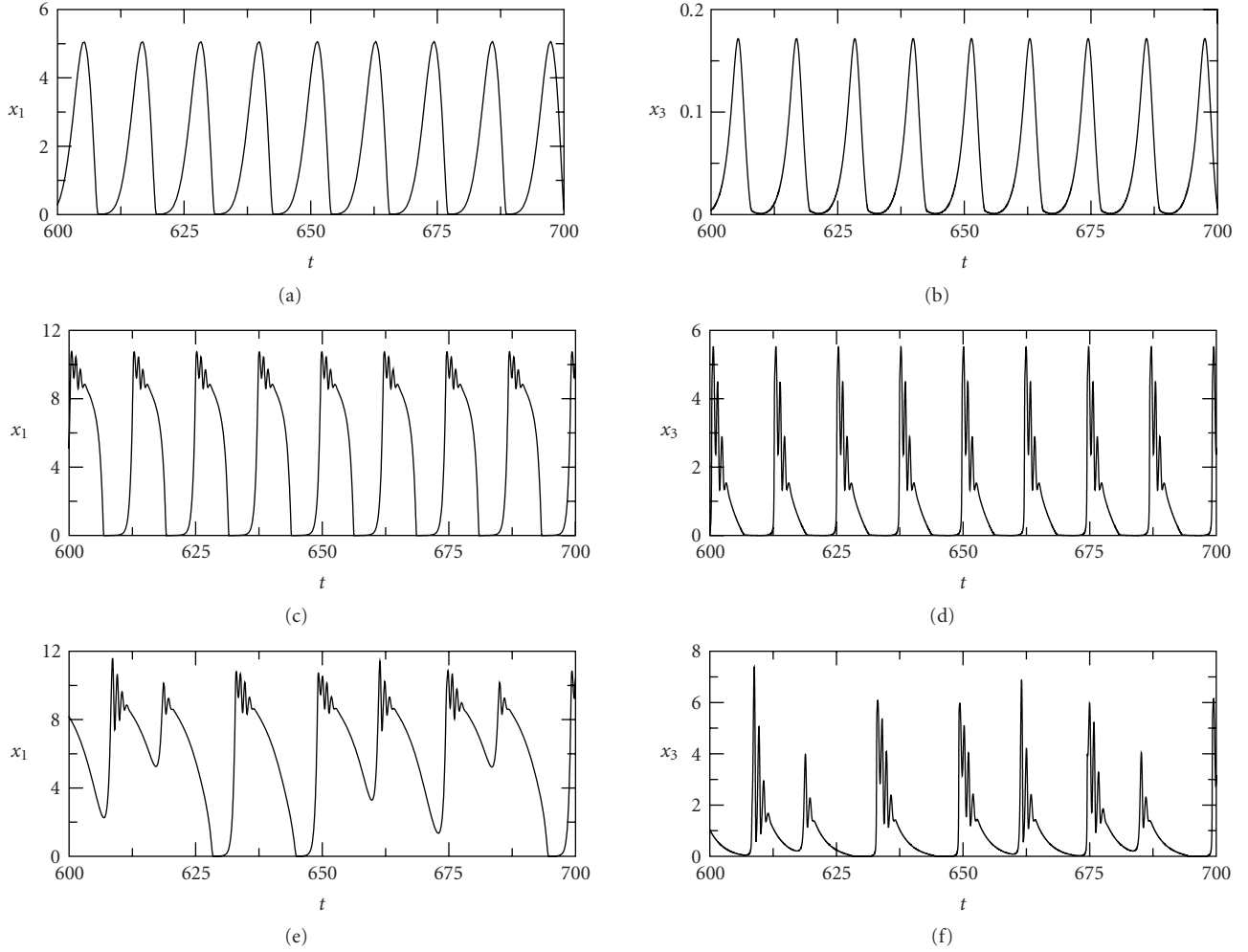


FIGURE 7: Kummer's calcium model [18]. x_1 is the activated $G\alpha$ subunits, x_3 is the concentration of free calcium in the cytosol. (a) and (b): periodic oscillations with $k_2 = 1.3$; (c) and (d): periodic bursting with $k_2 = 2.6$; (e) and (f): chaotic bursting with $k_2 = 2.7738$. The initial conditions: $x_1 = 0.01$, $x_2 = 0.01$, $x_3 = 0.01$, and $x_4 = 20$.

where x represents the concentration of cytoplasmic free calcium, A_{PM} , A_{ER} , and A_G are the total areas of plasma membrane, the ER membrane, and the gap-junctional connections, respectively, and C_C is the effective cytosolic volume (calcium "capacity"). The free calcium content of the whole cell is $z = x + (C_{ER}/C_C)y$, with y being the free calcium concentration in the ER. Here, we are concerned with a single cell so J_G is set to zero in what follows.

Stochastic simulation based on this model is done in [22]. Our purposes here are (1) to generalize Gillespie's stochastic simulation to the context of WR methodology, and (2) to show the feasibility of combining stochastic and deterministic simulation based on WR methodology. For these purposes, we reformulate the model using the variables x and y , instead of x and z , and also split the system into two components

Component 1:

$$\frac{dx}{dt} = \rho a^{(1)}(P, x, y) - \rho a^{(2)}(P, x). \quad (15)$$

Component 2:

$$\frac{dy}{dt} = \tau a^{(3)}(P, x) - \tau a^{(4)}(P, x, y), \quad (16)$$

where

$$\begin{aligned} a^{(1)} &= J_{in} + \alpha J_{rel}^+, \\ a^{(2)} &= J_{out} + \alpha (J_{rel}^- + J_{SERCA}), \\ a^{(3)} &= J_{rel}^- + J_{SERCA}, \\ a^{(4)} &= J_{rel}^+, \\ J_{in} &= \gamma_0 + \gamma_c \frac{P}{K_0 + P}, \\ J_{out} &= \gamma_4 \frac{x^2}{K_4^2 + x^2}, \\ J_{SERCA} &= \gamma_3 \frac{x^2}{K_3^2 + x^2}, \end{aligned}$$

$$\begin{aligned}
k_r(P, x) &= k_1 \left[\frac{d_2(d_1 + P)Px}{(d_1 + P)(d_a + x)(d_2(d_1 + P) + x(d_3 + P))} \right] + k_2, \\
\rho &= \frac{A_{PM}}{C_C}, \quad \alpha = \frac{A_{ER}}{A_{PM}}, \\
\beta &= \frac{C_{ER}}{C_C}, \quad \tau = \frac{A_{ER}}{C_{ER}}, \\
J_{rel} &= J_{rel}^+ - J_{rel}^-, \\
J_{rel}^+ &= k_r(P, x)y, \\
J_{rel}^- &= k_r(P, x)x
\end{aligned} \tag{17}$$

with P representing the concentration of inositol triphosphate (InsP3).

Based on the above model formulation, we can define stochastic processes in terms of variables X and Y , which are the numbers of calcium ions in the cell cytosol and the ER, respectively. Thus

$$x = \frac{X}{\Omega_C}, \quad y = \frac{Y}{\Omega_{ER}}, \tag{18}$$

where Ω_C and Ω_{ER} are the volumes of the cytosolic and ER cell compartments, respectively. Now, we can make three choices of WR simulation methods.

Method 1:

The deterministic WR. That is, we solve (15) and (16) for x and y separately and perform an iteration between the intermediate solutions to resolve the bidirectional coupling between two components.

Method 2:

The stochastic WR. Here, X and Y are treated as random variables and corresponding stochastic processes are defined by infinitesimal probabilities $\tilde{a}^{(i)} dt$. Thus for component 1,

$$\begin{aligned}
\tilde{a}^{(1)} &= \Omega_C \rho a^{(1)}: X \longrightarrow X + 1, \\
\tilde{a}^{(2)} &= \Omega_C \rho a^{(2)}: X \longrightarrow X - 1,
\end{aligned} \tag{19}$$

and for component 2

$$\begin{aligned}
\tilde{a}^{(3)} &= \Omega_{ER} \tau a^{(3)}: Y \longrightarrow Y + 1, \\
\tilde{a}^{(4)} &= \Omega_{ER} \tau a^{(4)}: Y \longrightarrow Y - 1.
\end{aligned} \tag{20}$$

Since the rate functions $\tilde{a}^{(1)}$ and $\tilde{a}^{(4)}$ are dependent on both X and Y , these two components are actually bidirectionally coupled and thus the WR method has to be used.

Method 3:

The combined deterministic and stochastic WR. Here, the state variable X for component 1 is treated as a random variable governed by the stochastic processes defined by (19), and the state variable y for component 2 is treated as a

continuous variable governed by (16) and is related to Y by (18). Such discrimination between components 1 and 2 is based on the fact that the concentration of calcium in the ER is much higher than that in the cytoplasm and hence is likely to be less influenced by stochastic fluctuations.

Figure 8 shows the computational output of these three methods. As expected, fluctuations appear in the stochastic simulation results, especially for the cytoplasmic-free calcium. Such a small size of fluctuations is generally observed in experimental studies. The magnitude of these fluctuations is controlled by Ω_C —the volume of the cytosolic compartment of the cell. When Ω_C is large enough, fluctuations become unobservable and the solution approaches the deterministic limit, in agreement with the theory and the results obtained in [22].

4.4. Model for Calcium Dynamics in a Cell Plate Mediated by Gap Junctions

In [17], the single cell model was extended to consider a cell pair linked by a gap junction, and the synchronization of heterogeneous cells was studied. Using our model linking strategy, we computed a multicellular version of Höfer's cell pair.

The model (see Figure 9) can be interpreted as representing a plate of hepatocytes spanning from the portal space to the hepatic venule in the liver, in agreement with the known anatomy. The liver plate is a sheet of cells approximately 10 cells high and 15–25 cells long [23]. For the purposes of computation, each cell in a vertical stack was taken to be equivalent and the spread of signal computed for a line of cells with the signal travelling in one dimension along the plate. The communication of cells in the plate is through gap junctions, and the cell-surface receptor density—the binding of hormone ligands to which activates the calcium pathway—is allowed to vary along the plate. The multicellular extension of Höfer's model is summarised as follows:

$$\begin{aligned}
\frac{dx_i}{dt} &= \rho_i(F_i - \alpha_i E_i) + \gamma_{i+1,i}(x_{i+1} - x_i) + \gamma_{i-1,i}(x_{i-1} - x_i), \\
\frac{dz_i}{dt} &= \rho_i F_i + \gamma_{i+1,i}(x_{i+1} - x_i) + \gamma_{i-1,i}(x_{i-1} - x_i), \\
F_i &= \nu_0 + \nu_C \frac{P_i}{K_0 + P_i} - \nu_4 \frac{x_i^2}{K_4^2 + x_i^2}, \\
E_i &= k_r(P_i, x_i) \beta_i^{-1} (z_i - (1 + \beta_i)x_i) - \nu_3 \frac{x_i^2}{K_3^2 + x_i^2}.
\end{aligned} \tag{21}$$

Here, each cell acts as a (nonlinear) calcium oscillator linearly coupled with its neighbours. The strength of the coupling is γ , representing the scaled gap junction permeability. The heterogeneity amongst cells comes from different sources: the variation of InsP3 levels (represented by P_i in the model) due to the density differences of receptors on different cell membranes, and the varying capacity of calcium stores between cells represented by the structure parameters

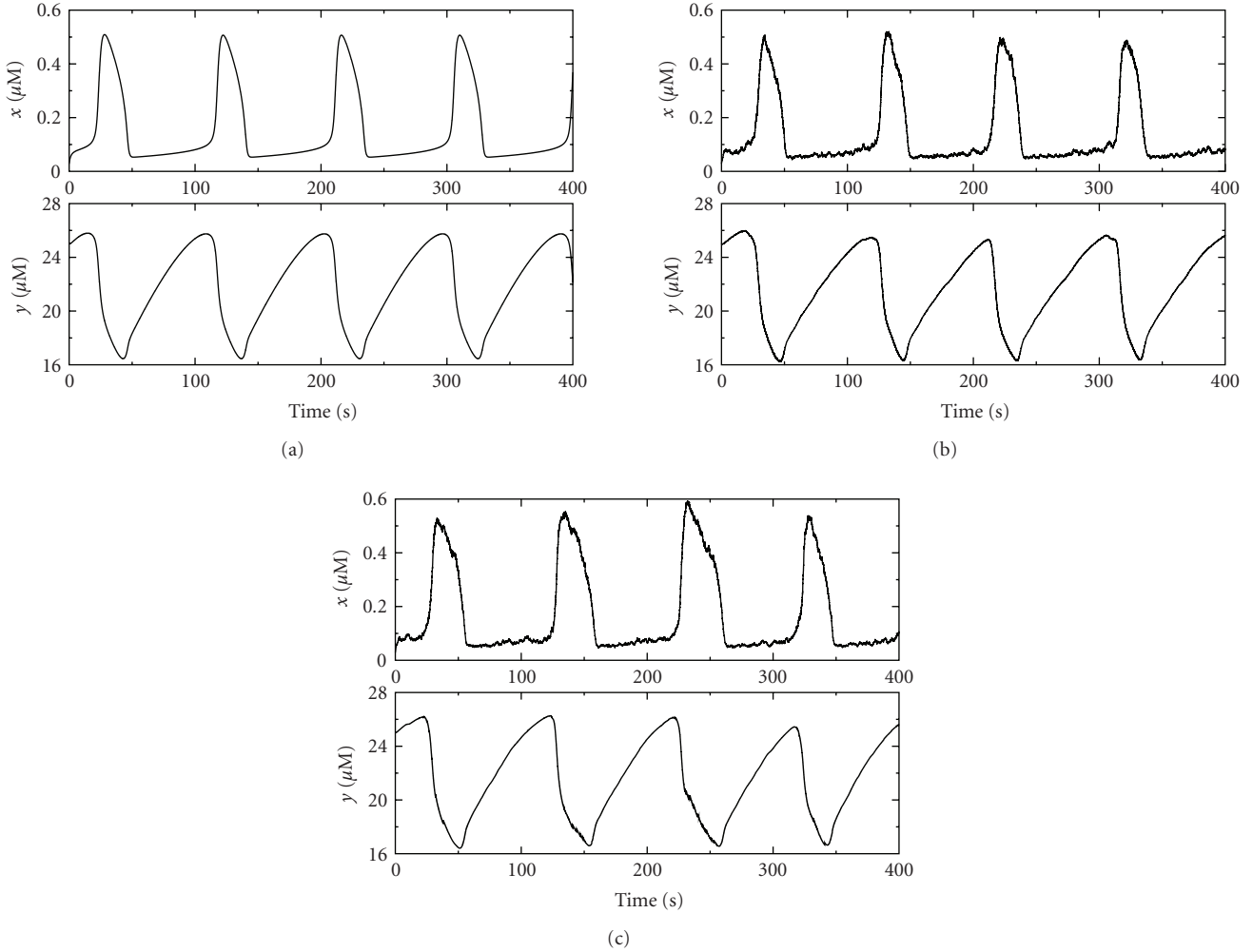


FIGURE 8: Calcium oscillations in the WR deterministic and/or stochastic versions of Höfer's model [17] for a single cell. (a) The solutions by the deterministic WR. (b) The solutions by the stochastic WR. (c) The solutions by the combined deterministic and stochastic WR. All the parameter values are taken from Höfer's original paper except for Ω_C and Ω_{ER} , which are set to be $5000 \mu\text{m}^3$ and $1000 \mu\text{m}^3$, respectively. The Initial conditions: rest state without agonist ($P = 0$).

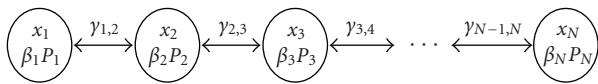


FIGURE 9: Model for the hepatic plate. A line of hepatic cells is bidirectionally coupled by gap junctions. Heterogeneity of the hepatocytes is controlled by the structure parameter β defined as the ratio of the effective volume of the ER to the effective cytosolic volume. The concentration of InsP3 is denoted by P . x_i denotes the concentration of cytoplasmic-free calcium in cell i . γ represents the coupling coefficient as defined by the scaled gap junction permeability. The model is based on Höfer's calcium oscillation model in hepatocytes [17].

β_i , defined as the ratio of the effective volume of the ER to the effective cytosolic volume. In the computations, InsP3 levels (P_i) among the cells are set up with a constant gradient, reflecting an experimentally observed density gradient of cell surface receptors between periportal and perivenous zones

[24]. The structure parameters β_i are treated as constants or random factors varying between 0.1 and 0.2 (these values are chosen from Höfer's estimation [17]).

Calcium oscillations are inherent to individual cells and the frequency and the shape of oscillations are determined by many factors, such as cell type, ligand and receptor densities, and so forth. In this particular model for hepatocytes, the oscillation frequency is governed by InsP3 levels and structural parameters. Therefore, for weak gap-junctional coupling (small values of γ), the calcium in each cell oscillates at its own inherent frequency, as indicated in Figures 10 and 11. However, when the permeability of gap junctions γ is increased, the computations show that the oscillations for individual cells become synchronized towards the frequency for the cell with the highest InsP3 level (Figures 10 and 11). This implies that the cell with the highest InsP3 level will direct the calcium waves in the liver lobule, in agreement with Höfer's analysis of a cell pair and also with experimental studies [24].

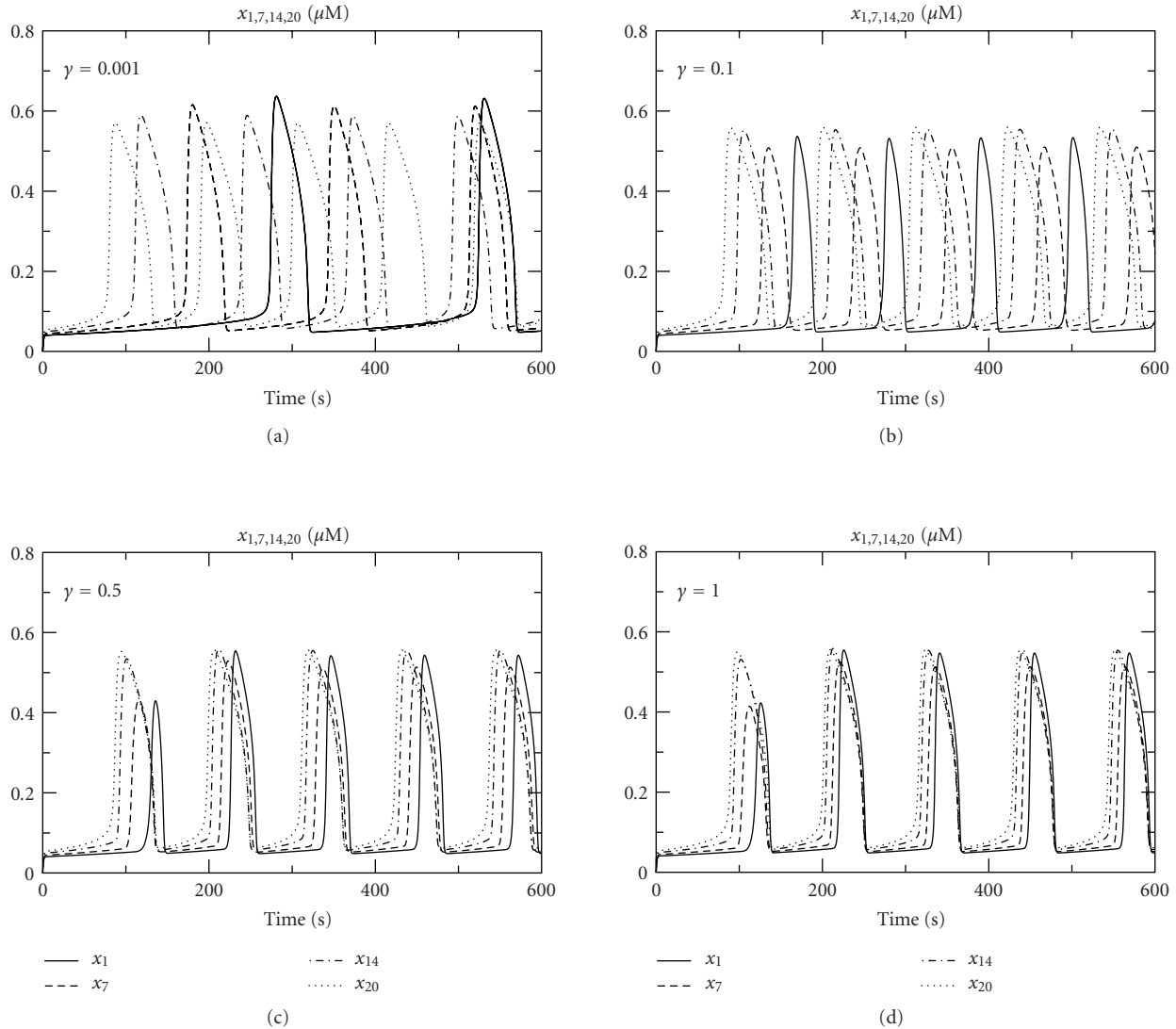


FIGURE 10: Synchronization of calcium oscillations in a plate of hepatocytes linked by gap junctions: $P_{i+1} = 1.05P_i$, $i = 1, 2, \dots, 19$, $P_1 = 1.5$, $\beta_i = 0.2$, $p_i = 0.02$, $\alpha_i = 2$, and $i = 1, 2, \dots, 20$. Initial conditions: rest state without agonist ($P = 0$).

When the Gauss-Seidel WR is applied to solve these bidirectionally coupled components, the convergence rate depends on the coupling strength γ . For a weak coupling, say $\gamma = 0.001, 2$, or 3 , iterations are sufficient to achieve convergence, but for an increasing γ the number of iterations increases.

5. Discussion and Conclusions

The main goal of systems biology and computational physiology is to formulate predictive models of integrated, functional biological systems which exploit the massive increase in low-level (usually molecular) detailed understanding and data that has accumulated over the past few decades. In parallel with biological developments, the history of mathematical and computational modelling of biological systems has largely concentrated on discrete, well-described,

single-scale, and isolatable subsystems. Taking the agenda of systems biology forward will require the integration of a number of these small-scale models of (comparatively) well-understood mechanisms into larger models in order to investigate the effects of dynamical interactions between the model components (e.g., for discrete multicellular systems such as organs). Further, several large and complicated models may need to be linked together to derive a biologically sensible model at a still larger scale (e.g., for whole-organism physiological systems).

A major challenge, therefore, is to take existing and new mathematical models of biological subsystems and integrate them together into new and larger systems. This is a difficult task to achieve for a number of reasons. First is the difficulty of delineating the appropriate biological modular structures, and identifying (or constructing) models of each such module. Second, the nature of biological interfaces between

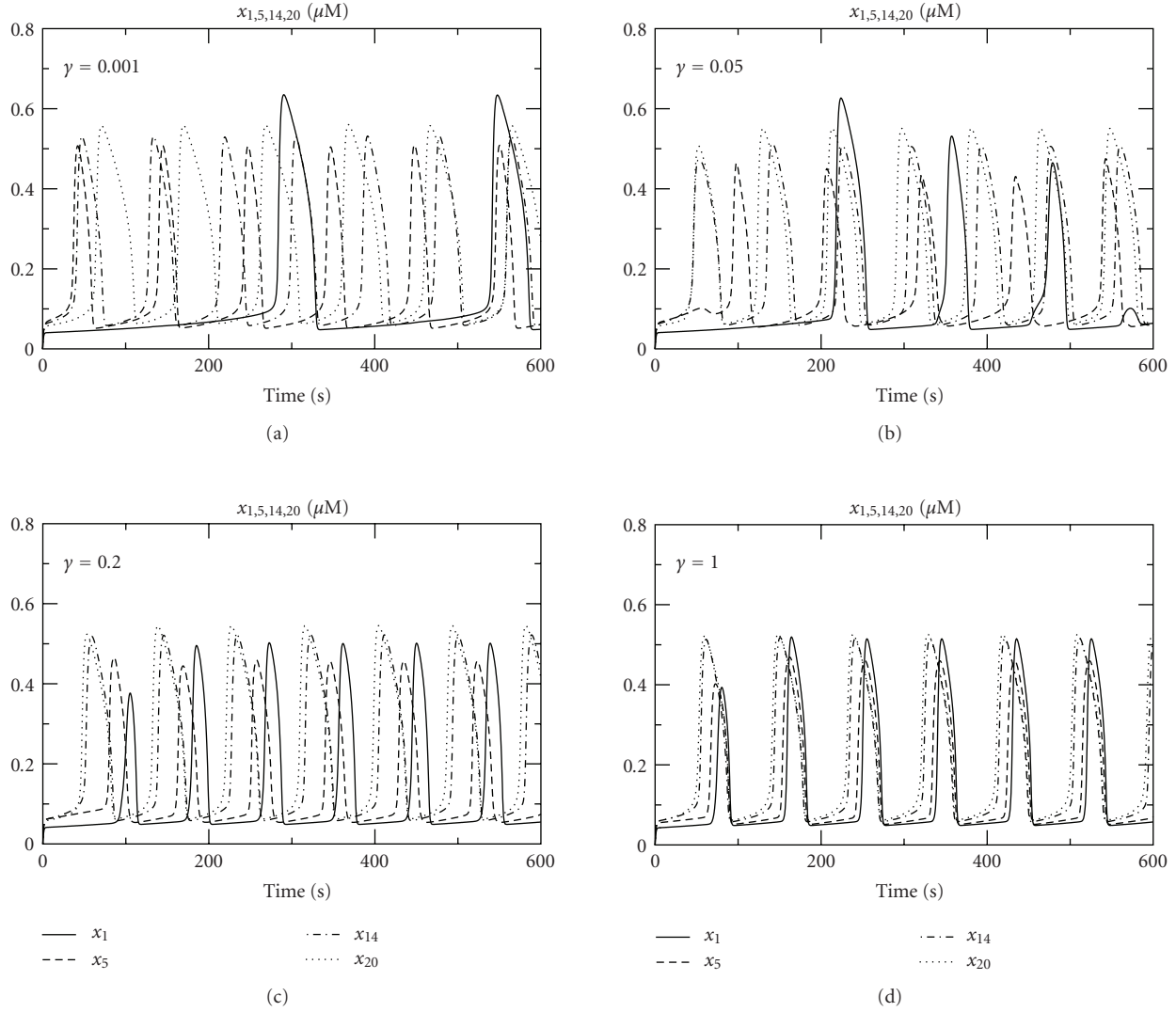


FIGURE 11: Synchronization of calcium oscillations in a plate of hepatocytes linked by gap junctions: $P_{i+1} = 1.05P_i$, $i = 1, 2, \dots, 19$, $P_1 = 1.5$, and β_i are randomly distributed between 0.1 and 0.2. In this particular computation, $\beta_1 = 0.2$, $\beta_5 = 0.1$, $\beta_{14} = 0.13$, and $\beta_{20} = 0.18$. Initial conditions: rest state without agonist ($P = 0$).

modules, and their representation as couplings amongst submodels must be formulated. A wide variation in the mathematical representation and software implementation of submodels often makes it impractical (or, at least, inefficient) to integrate submodels by simply reformulating them within a common computational paradigm. More fundamentally, it is impractical to construct a general reformulation framework to include all such potential variation. Because of this, any such special case reformulation will have limited generalizability. Third is designing an integration methodology that allows for efficient computation. Thus from both a practical point of view and a system level perspective, a general methodology that provides a means to capture the global solutions over time and space independently of the solution techniques related to individual submodels or components is required.

In this study, we propose waveform relaxation (WR) as a general methodology to integrate biological models to create large models of functional systems. The WR numerical method for parallel integration of systems of coupled ODEs is particularly useful because it allows the realisation of an integrated model without reformulating the whole problem, regardless of what formalisms were originally employed for each of the component models. We have demonstrated by case studies the multitime scale efficiency for bidirectionally coupled components, the convergence robustness, the flexibility, and the capability of tackling model heterogeneity of the WR method.

In principle, the WR strategy can be applied to a model system at any level of organisation, from the level of elementary chemical reactions to that of interactions in ecosystems. However, the efficiency of the method will

depend on both the network structure and the dynamics of the network, as we demonstrate in Section 4.1. This closely relates to the difficulty associated with defining the concept of modularity [25]. The main purpose of this paper is to provide a practical means—the WR method—for integrating model systems. Pragmatic, but informed, judgement concerning these issues will always be required.

Appendix

Stability of the Bidirectionally Coupled Harmonic Oscillator System

Let $y_i = \dot{x}_i/k_i$. Then we can rewrite the system defined by (12) as

$$\dot{\mathbf{X}} = \mathbf{D}\mathbf{X}, \quad \mathbf{X} = (X_1, \dots, X_N)^T, \quad X_i = (x_i, y_i)^T, \quad (\text{A.1})$$

where the Jacobian matrix \mathbf{D} is 2×2 block triangular:

$$\mathbf{D} = \begin{bmatrix} A_1 & \varepsilon B_2 & & & \\ B_1 & A_2 & \varepsilon B_3 & & \\ & \ddots & \ddots & \ddots & \\ & & B_{i-1} & A_i & \varepsilon B_{i+1} \\ & & & \ddots & \ddots & \ddots \\ & & & & B_{N-2} & A_{N-1} & \varepsilon B_N \\ & & & & & B_{N-1} & A_N \end{bmatrix} \quad (\text{A.2})$$

with

$$A_i = \begin{bmatrix} 0 & k_i \\ -k_i & 0 \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 & 0 \\ k_i & 0 \end{bmatrix}. \quad (\text{A.3})$$

We are required to find conditions under which \mathbf{D} has only complex eigenvalues, and hence that the system is decomposable into a system of stable harmonic oscillators. Thus if $P_N(\lambda) = |\lambda\mathbf{I} - \mathbf{D}|$ is the characteristic polynomial of \mathbf{D} , we consider conditions under which $P_N(\lambda) > 0$ for all real λ .

Consider the LU-decomposition of the tridiagonal block matrix $\lambda\mathbf{I} - \mathbf{D} = \mathbf{L}\mathbf{U}$, where

$$\mathbf{L} = \begin{bmatrix} L_1 & & & & \\ -B_1 & L_2 & & & \\ & \ddots & \ddots & & \\ & & -B_{i-1} & L_i & \\ & & & \ddots & \ddots \\ & & & & -B_{N-2} & L_{N-1} \\ & & & & & -B_{N-1} & L_N \end{bmatrix}, \quad (\text{A.4})$$

$$\mathbf{U} = \begin{bmatrix} I_2 & U_2 & & & \\ & I_2 & U_3 & & \\ & & \ddots & \ddots & \\ & & & I_2 & U_i \\ & & & & \ddots & \ddots \\ & & & & & I_2 & U_N \\ & & & & & & I_2 \end{bmatrix}.$$

Then

$$L_1 = \lambda I_2 - A_1, \quad (\text{A.5})$$

$$L_i = \lambda I_2 - A_i - \varepsilon B_{i-1} L_{i-1}^{-1} B_i, \quad 2 \leq i \leq N.$$

It now follows easily from (A.3) and (A.5) that

$$P_N(\lambda) = |L||U| = \prod_{i=1}^N |L_i|, \quad (\text{A.6})$$

$$|L_1(\lambda)| = \lambda^2 + k_1^2,$$

$$|L_i(\lambda)| = \lambda^2 + k_i^2 \left(1 - \frac{\varepsilon k_{i-1}^2}{|L_{i-1}(\lambda)|} \right), \quad 2 \leq i \leq N. \quad (\text{A.7})$$

From (A.6), we have $P_i(\lambda) = |L_i(\lambda)|P_{i-1}(\lambda)$ for $i \geq 1$, where we set $P_0(\lambda) = 1$. It follows easily by induction from (A.7) that $|L_i(\lambda)| \geq |L_i(0)|$ for all real λ and all $i \geq 1$.

Set $R_0(\varepsilon) = 1$ and

$$R_i(\varepsilon) = \frac{1}{\prod_{j=1}^i k_j^2} P_i(0), \quad (\text{A.8})$$

for $i \geq 1$. Then $P_i(0) = |L_i(0)|P_{i-1}(0)$ implies that

$$|L_i(0)| = \frac{k_i^2 R_i(\varepsilon)}{R_{i-1}(\varepsilon)}. \quad (\text{A.9})$$

It follows that $|L_i(\lambda)| > 0$ for all real λ and $i \geq 1$ if and only if $R_i(\varepsilon) > 0$ for each $i \geq 0$.

Clearly, $P_0(\lambda) = P_0(0) > 0$ for all real λ . Assume inductively that $P_j(\lambda) > 0$ for all real λ and $0 \leq j < i$. Then $P_i(\lambda) = |L_i(\lambda)|P_{i-1}(\lambda)$ implies that

$$P_i(\lambda) > 0, \quad \text{for all real } \lambda \iff |L_i(\lambda)| > 0, \quad (\text{A.10})$$

$$\text{for all real } \lambda \iff R_i(\varepsilon) > 0.$$

Thus setting

$$\Omega_i = \{\varepsilon \in \mathbb{R} : R_i(\varepsilon) > 0\}, \quad (\text{A.11})$$

it follows by induction that $P_i(\lambda) > 0$ for all real λ , and all $0 \leq i \leq N$ if and only if

$$\varepsilon \in \Gamma_N = \bigcap_{i=1}^N \Omega_i. \quad (\text{A.12})$$

We now obtain the following.

Theorem A.1. *For $k_i > 0$, $i = 1, 2, \dots, N$, and real ε , the system (A.1) consists of N stable harmonic oscillators if and only if $\varepsilon \in \Gamma_N = (-\infty, \varepsilon_N^*)$, where*

$$\varepsilon_N^* = \frac{1}{4} \left[1 + \tan^2 \left(\frac{\pi}{N+1} \right) \right]. \quad (\text{A.13})$$

Thus ε_N^* is monotonically decreasing in N , and $\lim_{N \rightarrow \infty} \varepsilon_N^* = 1/4$.

Proof. It remains to identify the set $\Gamma_N \subseteq \mathbb{R}$. From (A.7), we have

$$|L_i(0)| = k_i^2 \left\{ 1 - \varepsilon \frac{k_{i-1}^2}{|L_{i-1}(0)|} \right\}. \quad (\text{A.14})$$

Substituting from (A.9) gives

$$\frac{k_i^2 R_i(\varepsilon)}{R_{i-1}(\varepsilon)} = k_i^2 \left\{ 1 - \varepsilon \frac{R_{i-2}(\varepsilon)}{R_{i-1}(\varepsilon)} \right\}, \quad (\text{A.15})$$

for $i \geq 2$, and hence

$$R_i(\varepsilon) = R_{i-1}(\varepsilon) - \varepsilon R_{i-2}(\varepsilon). \quad (\text{A.16})$$

The solution of the difference equation (A.16) with initial condition $R_0(\varepsilon) = R_1(\varepsilon) = 1$ is

$$R_i(\varepsilon) = \begin{cases} \left(\frac{1-r_-}{r_+ - r_-} \right) r_+^i - \left(\frac{1-r_+}{r_+ - r_-} \right) r_-^i, & \varepsilon \neq \frac{1}{4}, \\ (i+1)2^{-i}, & \varepsilon = \frac{1}{4}, \end{cases} \quad (\text{A.17})$$

where r_{\pm} are the roots of the quadratic $r^2 - r + \varepsilon = 0$. That is, $r_{\pm} = (1/2)(1 \pm \sqrt{1-4\varepsilon})$. Clearly, these are real and distinct for $\varepsilon < 1/4$ and complex for $\varepsilon > 1/4$.

Setting $\varepsilon = 1/4 - \delta^2$, for $\delta \geq 0$ transforms (A.17) to

$$R_i(\varepsilon) = \begin{cases} \frac{1}{2\delta} [(1+2\delta)^{i+1} - (1-2\delta)^{i+1}], & \delta > 0, \\ (i+1)2^{-i}, & \delta = 0, \end{cases} \quad (\text{A.18})$$

and we conclude that $R_i(\varepsilon) > 0$, for all $i \geq 0$ and $\varepsilon \leq 1/4$.

For $\varepsilon > 1/4$, set $\varepsilon = 1/4 + \delta^2$ with $\delta > 0$. Then (A.17) gives

$$R_k(\varepsilon) = \frac{1}{2\delta i} [(1+2\delta i)^{k+1} - (1-2\delta i)^{k+1}]$$

$$= \frac{(1+4\delta^2)^{(k+1)/2}}{2\delta} \sin[(k+1)\varphi(\delta)], \quad (\text{A.19})$$

where $\varphi(\delta) = \tan^{-1}(2\delta)$. Thus $R_k(\varepsilon) = 0$ if and only if $(k+1)\varphi(\delta) = m\pi$ for positive integer $m < (1/2)(k+1)$. This gives a finite set of possible values $\delta_m = (1/2)\tan[m\pi/(k+1)]$, for $m = 1, 2, \dots, [k/2]$, and hence a corresponding finite set of real roots of $R_k(\varepsilon) = 0$:

$$\varepsilon_k^{(m)} = \frac{1}{4} \left[1 + \tan^2 \left(\frac{m\pi}{k+1} \right) \right], \quad m = 1, 2, \dots, \left[\frac{k}{2} \right]. \quad (\text{A.20})$$

Clearly, $\varepsilon_k^{(1)}$ is the smallest of these roots. Also, $\varepsilon_k^{(1)}$ is decreasing in k , and hence $\varepsilon_N^{(1)} \leq \varepsilon_k^{(1)}$, for $1 \leq k \leq N$. It, therefore, follows that $R_k(\varepsilon) > 0$, for $1 \leq k \leq N$ and all $\varepsilon < \varepsilon_N^* = \varepsilon_N^{(1)}$, which gives (A.13) and completes the proof of the theorem. \square

Acknowledgments

The authors would like to thank the DTI for making this work possible through the award of a Beacon Project (Lead PI Professor Anne Warner). Linzhong Li was supported by the Beacon award. They thank Anne Warner for useful discussion and comment on the manuscript and their colleagues in CoMPLEX for additional advice and comment.

References

- [1] F. C. Boogerd, F. J. Bruggeman, J. S. Hofmeyr, and H. V. Westerhoff, Eds., *Systems Biology: Philosophical Foundations*, Elsevier, Amsterdam, The Netherlands, 2007.
- [2] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic*, Academic Press, San Diego, Calif, USA, 2000.
- [3] K. Takahashi, K. Kaaizu, B. Hu, and M. Tomita, "A multi-algorithm, multi-timescale method for cell simulation," *Bioinformatics*, vol. 20, no. 4, pp. 538–546, 2004.
- [4] A. M. Uhrmacher, "Concepts of object- and agent-oriented simulation," *Transactions of the Society for Computer Simulation*, vol. 14, no. 2, pp. 59–67, 1997.
- [5] A. R. Asthagiri and D. A. Lauffenburger, "Bioengineering models of cell signaling," *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 31–53, 2000.
- [6] K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations*, Oxford University Press, Oxford, UK, 1995.

- [7] H. Schurz and K. R. Schneider, "Waveform relaxation methods for ordinary stochastic differential equations," *International Journal of Numerical Analysis and Modeling*, vol. 3, no. 2, pp. 232–254, 2005.
- [8] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, no. 4, pp. 403–434, 1976.
- [9] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *Journal of Physical Chemistry A*, vol. 104, no. 9, pp. 1876–1889, 2000.
- [10] H. H. McAdams and A. Arkin, "Stochastic mechanisms in gene expression," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 94, no. 3, pp. 814–819, 1997.
- [11] H. P. Grimm, A. B. Verkhovsky, A. Mogilner, and J.-J. Meister, "Analysis of actin dynamics at the leading edge of crawling cells: implications for the shape of keratocyte lamellipodia," *European Biophysics Journal*, vol. 32, no. 6, pp. 563–577, 2003.
- [12] National Research Council, *Mathematics and 21st Century Biology*, The National Academies Press, Washington, DC, USA, 2005.
- [13] Ch. Engstler and Ch. Lubich, "Multirate extrapolation methods for differential equations with different time scales," *Computing*, vol. 58, no. 2, pp. 173–185, 1997.
- [14] B. Pohl, "On the convergence of the discretized multi-splitting waveform relaxation algorithm," *Applied Numerical Mathematics*, vol. 11, no. 1–3, pp. 251–258, 1993.
- [15] G. Engeln-Mullges and F. Uhlig, *Numerical Algorithms with C*, Springer, Berlin, Germany, 1996.
- [16] C. Carlin and C. Vachoux, "On partitioning for waveform relaxation time-domain analysis of VLSI circuits," in *Proceedings of the International Conference on Circuits and Systems (ISCAS '84)*, Montreal, Canada, May 1984.
- [17] T. Höfer, "Model of intercellular calcium oscillations in hepatocytes: synchronization of heterogeneous cells," *Biophysical Journal*, vol. 77, no. 3, pp. 1244–1256, 1999.
- [18] U. Kummer, L. F. Olsen, C. J. Dixon, A. K. Green, E. Bornberg-Bauer, and G. Baier, "Switching from simple to complex oscillations in calcium signaling," *Biophysical Journal*, vol. 79, no. 3, pp. 1188–1195, 2000.
- [19] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.
- [20] P. J. Prince and J. R. Dormand, "High order embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 7, no. 1, pp. 67–75, 1981.
- [21] D. A. Lauffenburger and J. J. Linderman, *Receptors: Models for Binding, Trafficking, and Signalling*, Oxford University Press, Oxford, UK, 1993.
- [22] M. E. Gracheva, R. Toral, and J. D. Gunton, "Stochastic effects in intercellular calcium spiking in hepatocytes," *Journal of Theoretical Biology*, vol. 212, no. 1, pp. 111–125, 2001.
- [23] J. Gumucio, B. M. Bilir, R. H. Moseley, and C. M. Berkowitz, "The biology of the liver cell plate," in *The Liver Biology and Pathobiology*, I. M. Arias, J. L. Boyer, N. Fausto, W. B. Jakoby, D. Schachter, and D. A. Shafritz, Eds., pp. 1143–1163, Raven Press, New York, NY, USA, 3rd edition, 1994.
- [24] C. Clair, D. Tran, S. Boucherie, M. Claret, T. Tordjmann, and L. Combettes, "Hormone receptor gradients supporting directional Ca^{2+} signals: direct evidence in rat hepatocytes," *Journal of Hepatology*, vol. 39, no. 4, pp. 489–495, 2003.
- [25] Z. Szallasi, V. Periwal, and J. Stelling, "On modules and modularity," in *System Modelling in Cellular Biology*, Z. Szallasi, J. Stelling, and V. Periwal, Eds., pp. 41–50, The MIT Press, London, UK, 2006.